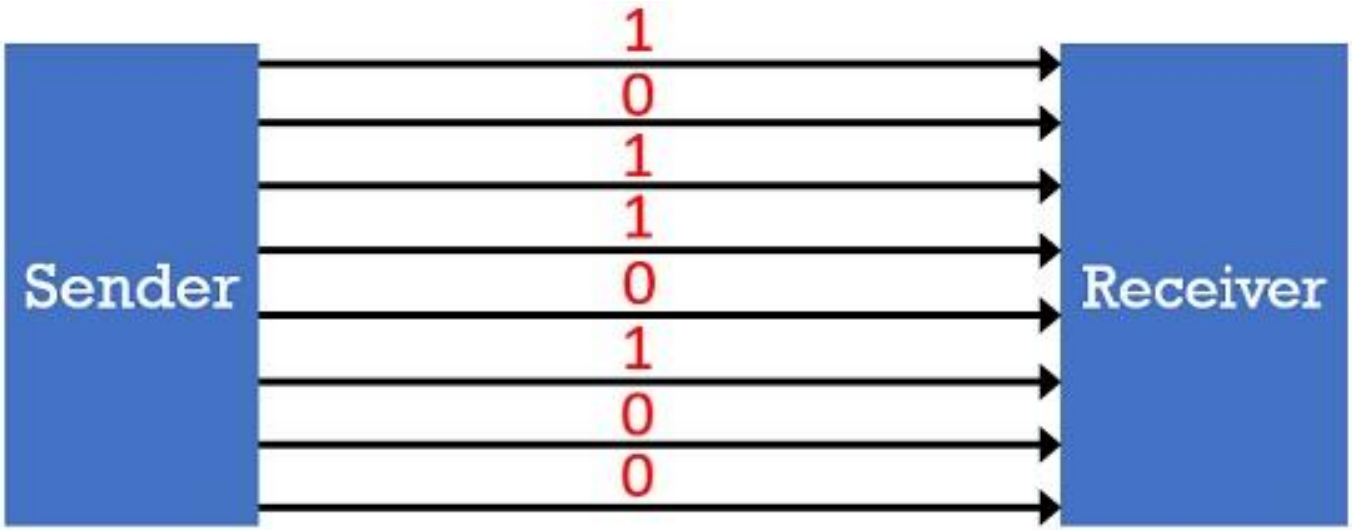
Serial And Parallel Transmission

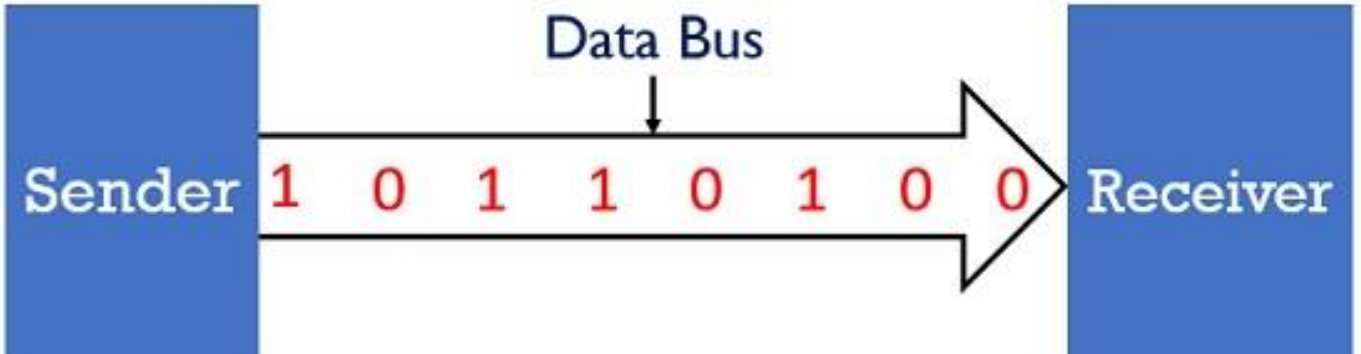
There are two methods used to **transmit** data between digital devices: **serial transmission** and **parallel transmission**. **Serial** data **transmission** sends data bits one after another over a single wire. **Parallel** data **transmission** sends multiple data bits at the same time over multiple wires.

Normally Peripheral devices uses Serial data transmission. On the other hand, computer uses parallel data transmission concept.



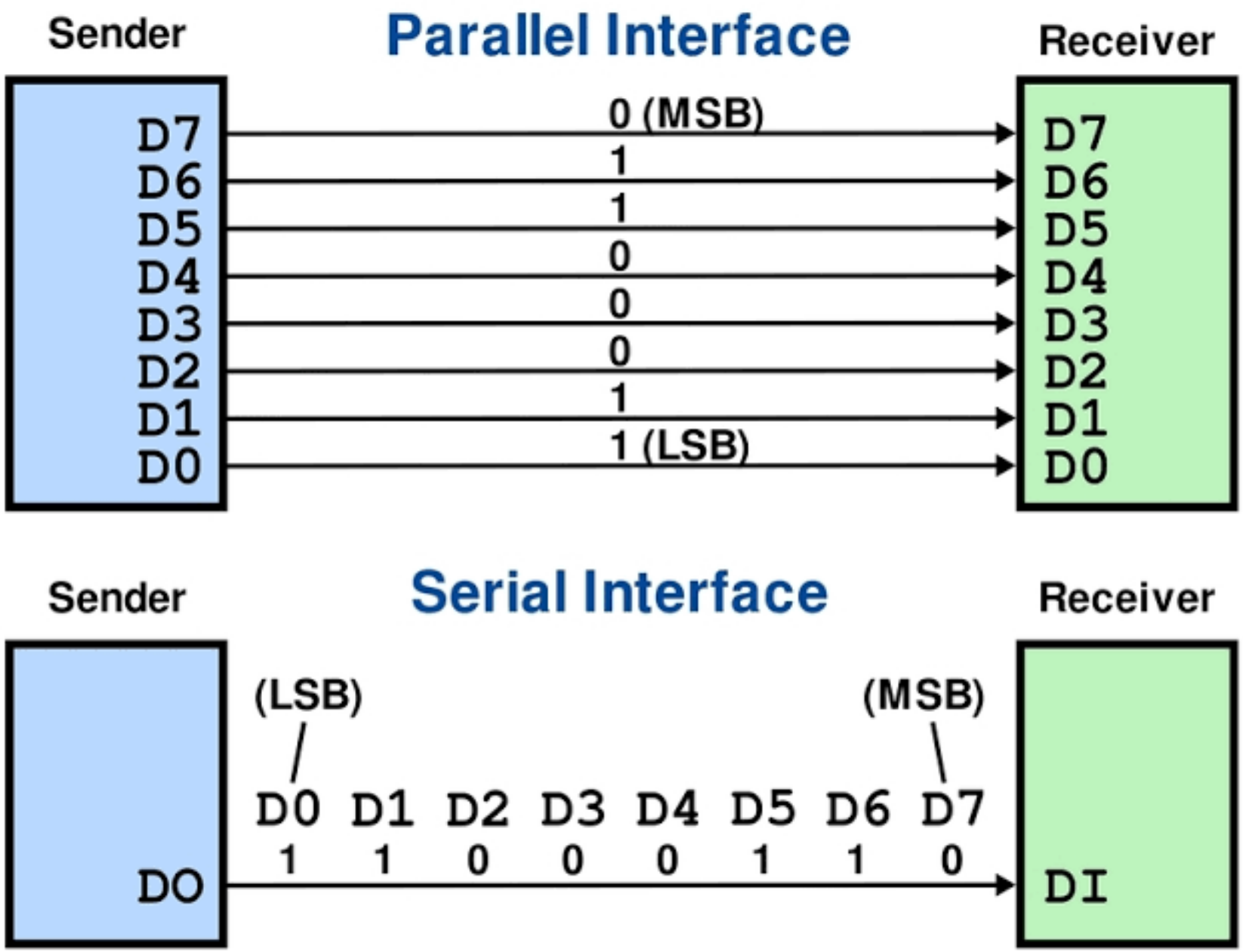
Parallel Communication

Circuit Globe



Serial Communication

Circuit Globe


Sender
Parallel Interface
Receiver
D7
D6
D5
D4
D3
D2
D1
D0
0 (MSB)
1
1
0
0
0
1
1 (LSB)
D7
D6
D5
D4
D3
D2
D1
D0
Sender
Serial Interface
Receiver
(LSB)
(MSB)
D0 D1 D2 D3 D4 D5 D6 D7
1 1 0 0 0 1 1 0
DO
DI

Serial Interface

•Considered to be one of the most basic external connections to a computer, the **serial port** has been an integral part of most computers for more than 20 years. Although many of the newer systems have done away with the serial port completely in favor of USB connections, most modems still use the serial port, as do some printers, PDAs(Personal Device Assistant) and digital cameras.

•Essentially, serial ports provide a standard connector and protocol to let you attach devices, such as modems, to your computer.

Serial Interface

The name "serial" comes from the fact that a serial port "serializes" data. That is, it takes a byte of data and transmits the 8 bits in the byte one at a time.

The advantage is that a serial port needs only one wire to transmit the 8 bits (while a parallel port needs 8). So to send data in long distance it may be converted in serial form. Serial ports lower cable costs and make cables smaller.

The disadvantage is that it takes 8 times longer to transmit the data than it would if there were 8 wires.

Serial Interface

Serial ports, also called **communication (COM) ports**, are **bi-directional**.

Bi-directional communication allows each device to receive data as well as transmit it. Serial devices use different pins to receive and transmit data -- using the same pins would limit communication to **half-duplex**, meaning that information could only travel in one direction at a time.

Using different pins allows for **full-duplex** communication, in which information can travel in both directions at once.

Serial Interface

•Data is moved in parallel within a computer. To interface a computer with serial data lines, the data must be converted to and from serial form.

•A parallel-in-serial-out shift register and a serial-in-parallel-out shift register can be used to do this.

Parallel-Serial Conversion

- For Transmission, parallel data word is loaded into the shift register.
- A pulse on the clock input causes the data to be shifted.
- For an n-bit data word n clock pulses will output the word in serial form.

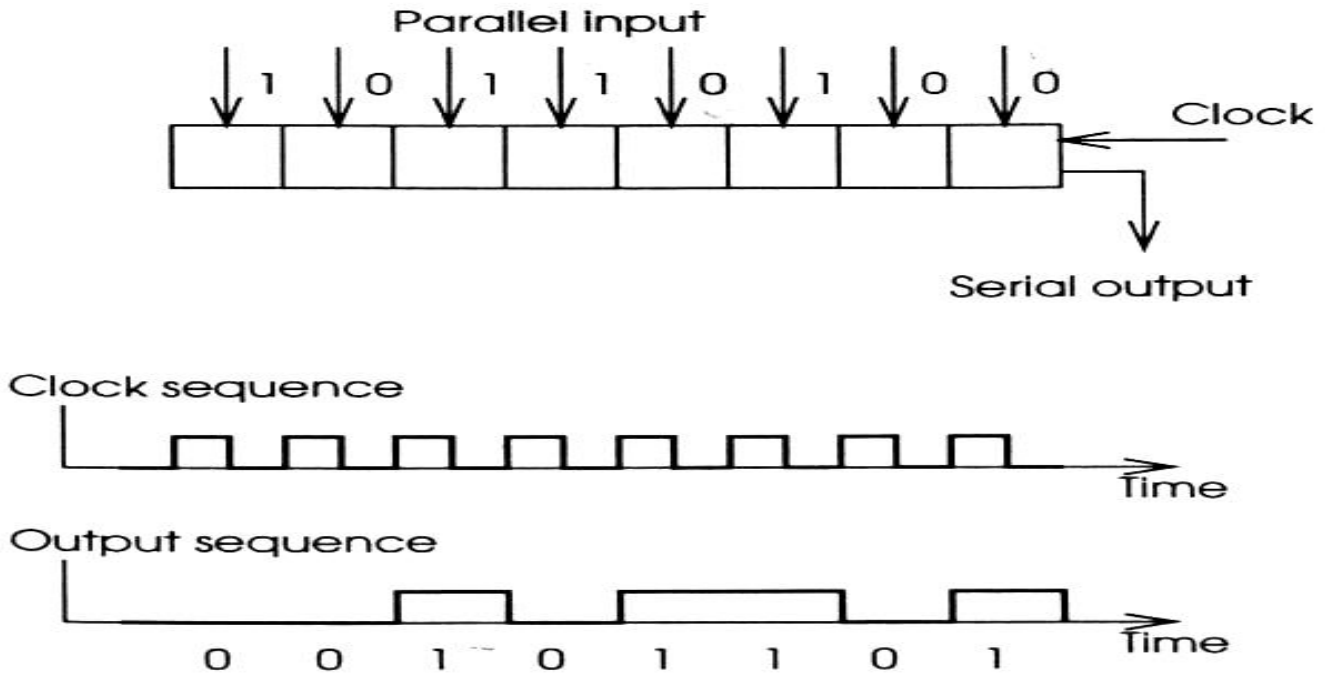


Figure 2.14 *Parallel-to-serial conversion using a shift register*

Serial-Parallel Conversion

- **Reception of the serial data is performed by another shift register, in a serial-to-parallel convertor.**
- **Asequence of n clock pulses causes the input to propagate along the shift register until it is all available in parallel.**
- **The first bit to arrive is shifted all the way through the shift register and appears at the right hand end.**

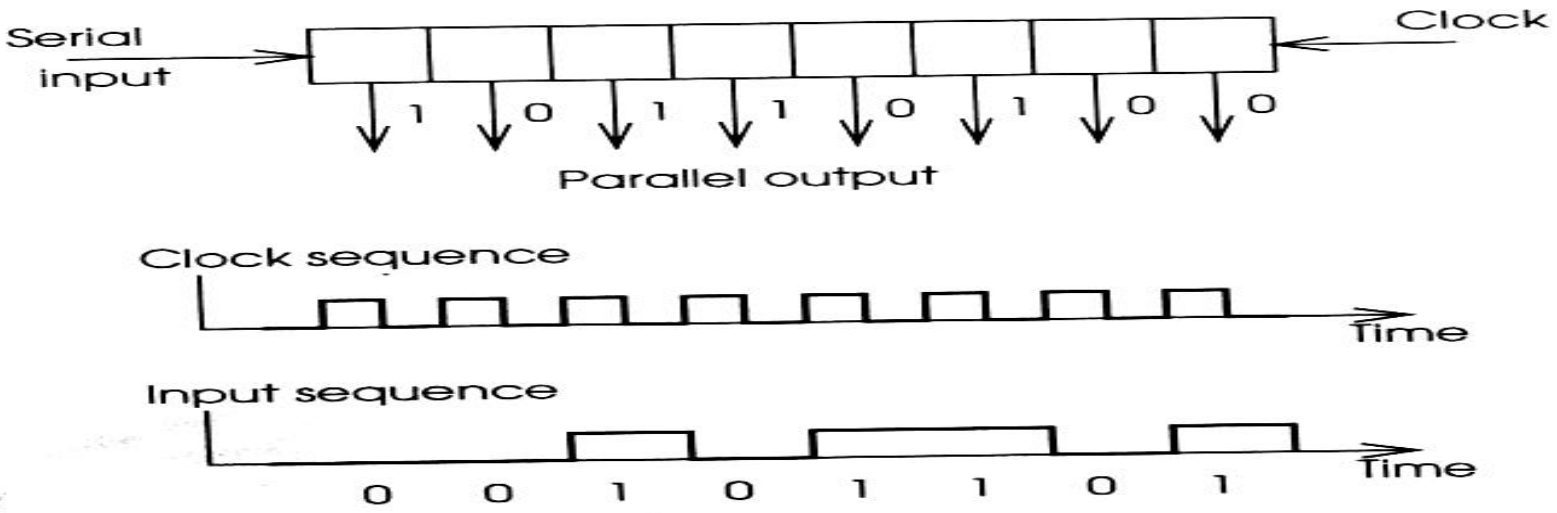
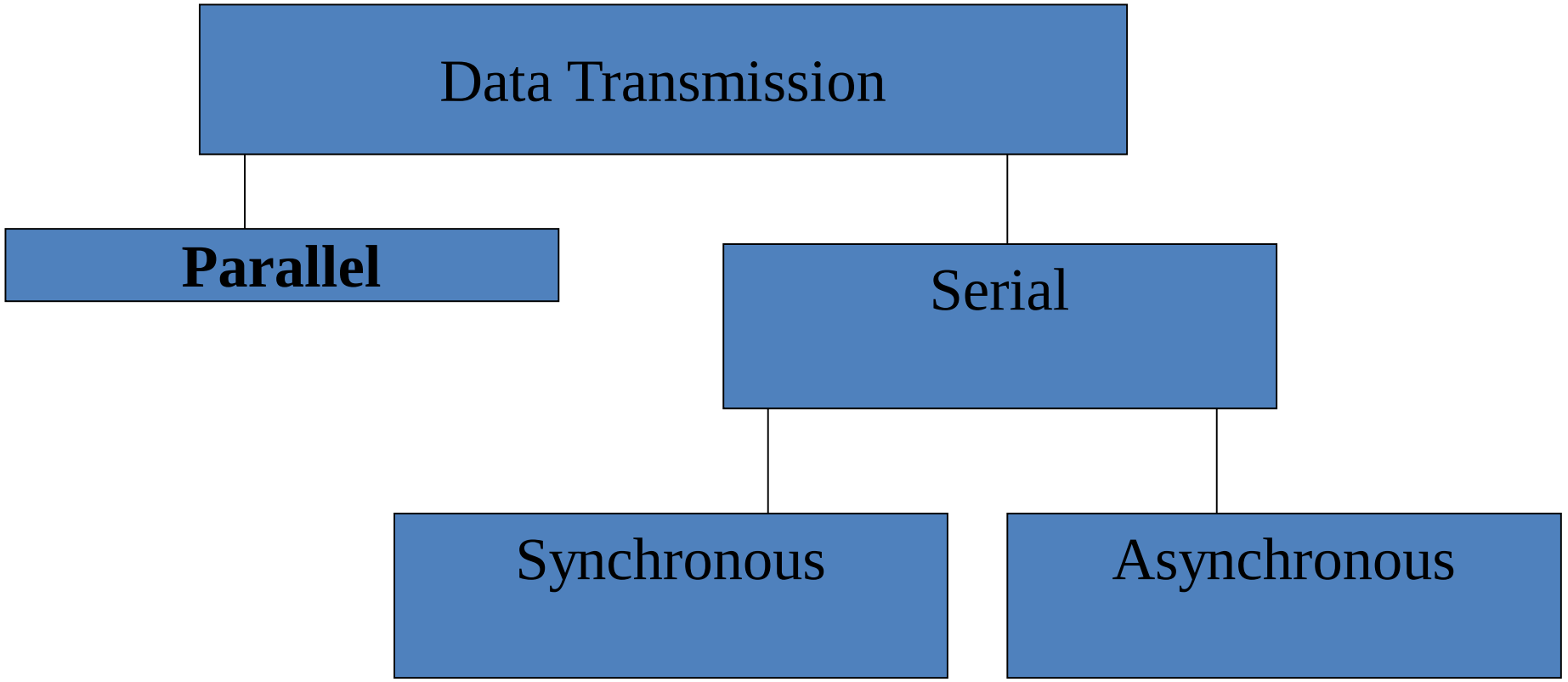


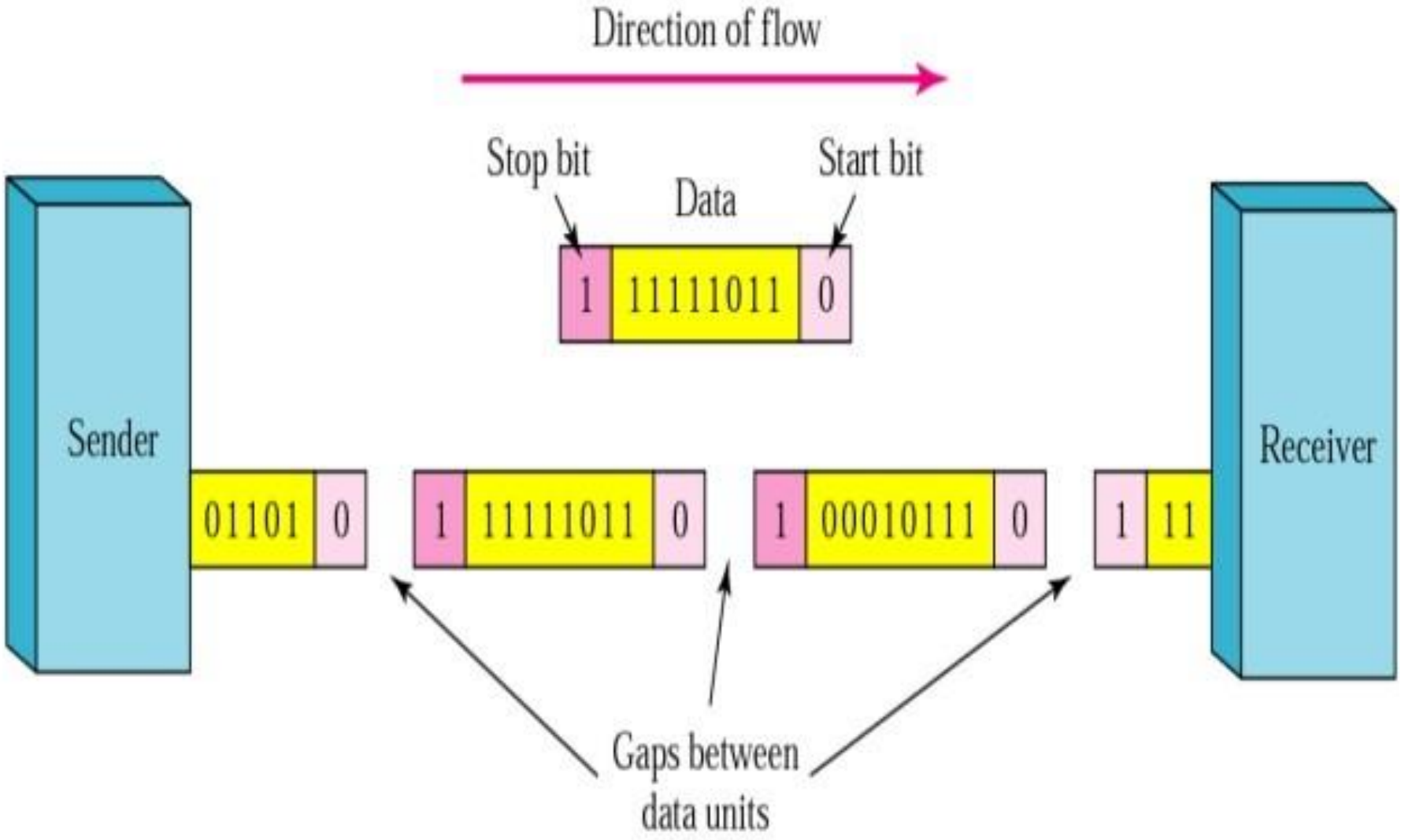
Figure 2.15 *Serial-to-parallel conversion using a shift register*

Data Transmission Tree



Serial data can be sent synchronously or asynchronously.

Asynchronous Transmission



Asynchronous Transmission

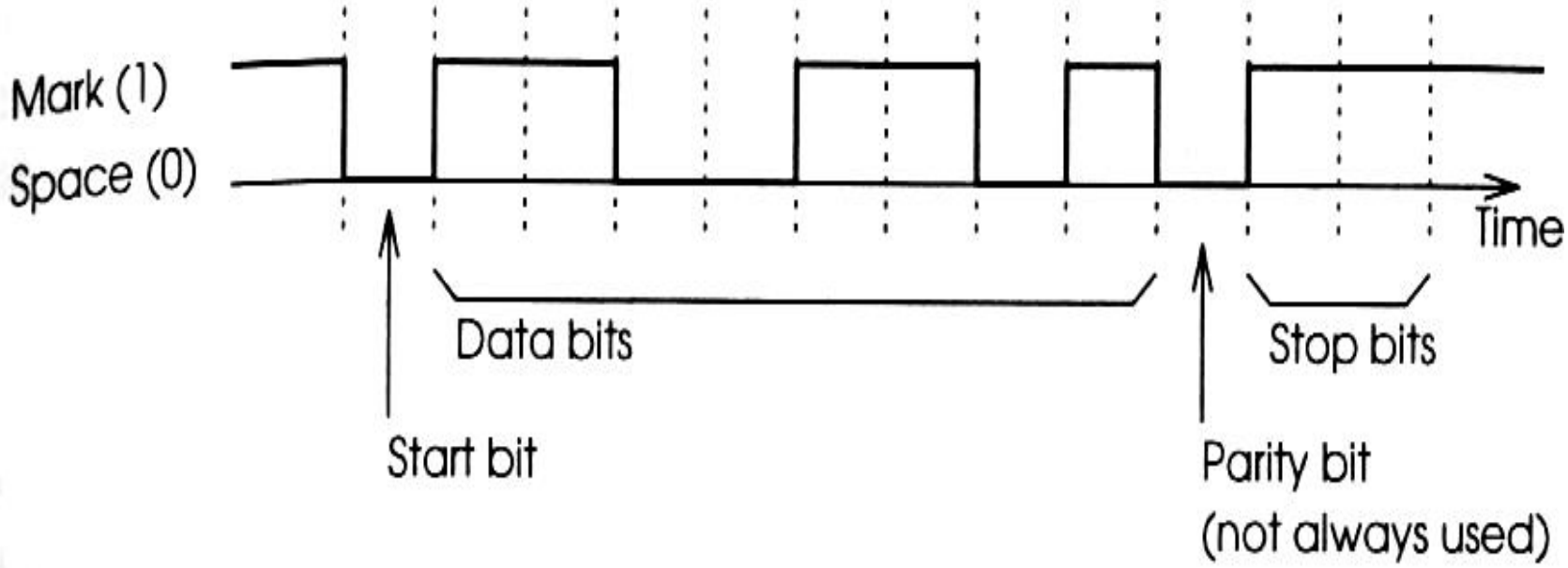


Figure 2.16 *Asynchronous character format*

Definitions

- **Start Bit**
  - Signals the beginning of the data word
  - A low bit after a series of high bits
- **Data Bits**
  - The meat of the transmission
  - Usually 7 or 8 bits

Definitions Continued

- **Parity Bit**
  - An error check bit placed after the data bits
  - Can be high or low depending on whether odd parity or even parity is specified
- **Stop Bit/s**
  - One or two high bits that signal the end or the data word
- **Data Word**
  - Start Bit, Data Bits, Parity Bit, & Stop Bit/s

In **Synchronous transmission** data is sent in the form block by block. This transmission is the full duplex type. Between sender and receiver the synchronization is compulsory. In Synchronous transmission, There is no gap present between data. It is more efficient and more reliable than asynchronous transmission to transfer the large amount of data.

On other hand in **Asynchronous transmission** data is **transmitted** in the form of byte or character. This transmission is the half duplex type transmission. In this transmission start bits and stop bits are added with data. It does not require synchronization.

SL. No	Synchronous Transmission	Asynchronous Transmission
01	In Synchronous transmission, Data is sent in form of blocks or frames.	In asynchronous transmission, Data is sent in form of byte or character.
02	Synchronous transmission is fast.	Asynchronous transmission is slow.
03	Synchronous transmission is costly.	Asynchronous transmission less costly.
04	In Synchronous transmission, There is no gap present between data.	In asynchronous transmission, There is present gap between data.
05	Efficient use of transmission line is done in synchronous transmission.	While in asynchronous transmission, transmission line remains empty during gap in character transmission.
06	Synchronous transmission needs precisely synchronized clocks for the information of new bytes.	Asynchronous transmission have no need of synchronized clocks as parity bit is used in this transmission for information of new bytes.
07	It is full duplex mode communication	It is half duplex mode communication.

FYI Term: “UART”

“…

- **Universal**
- **Asynchronous**
- **Receiver-**
- **Transmitter**

a computer component that handles asynchronous serial communication.”

www.webopedia.com

BAUD RATE

$\neq$

BIT RATE

Bit Rate and Baud Rate

Bit rate is a measure of the number of data bits (that's 0's and 1's) transmitted in one second.

Baud Rate(signal unit/ sec)

- The number of signal units per second
- Less than or equal to the bit rate

Synchronous Transmission

- Data are sent continuously.
- There is no start bit to indicate where a character begins. So this must be achieved by special sequence of data known as protocol.
- One such protocol HDLC – High Level Data Link Control format is shown below:

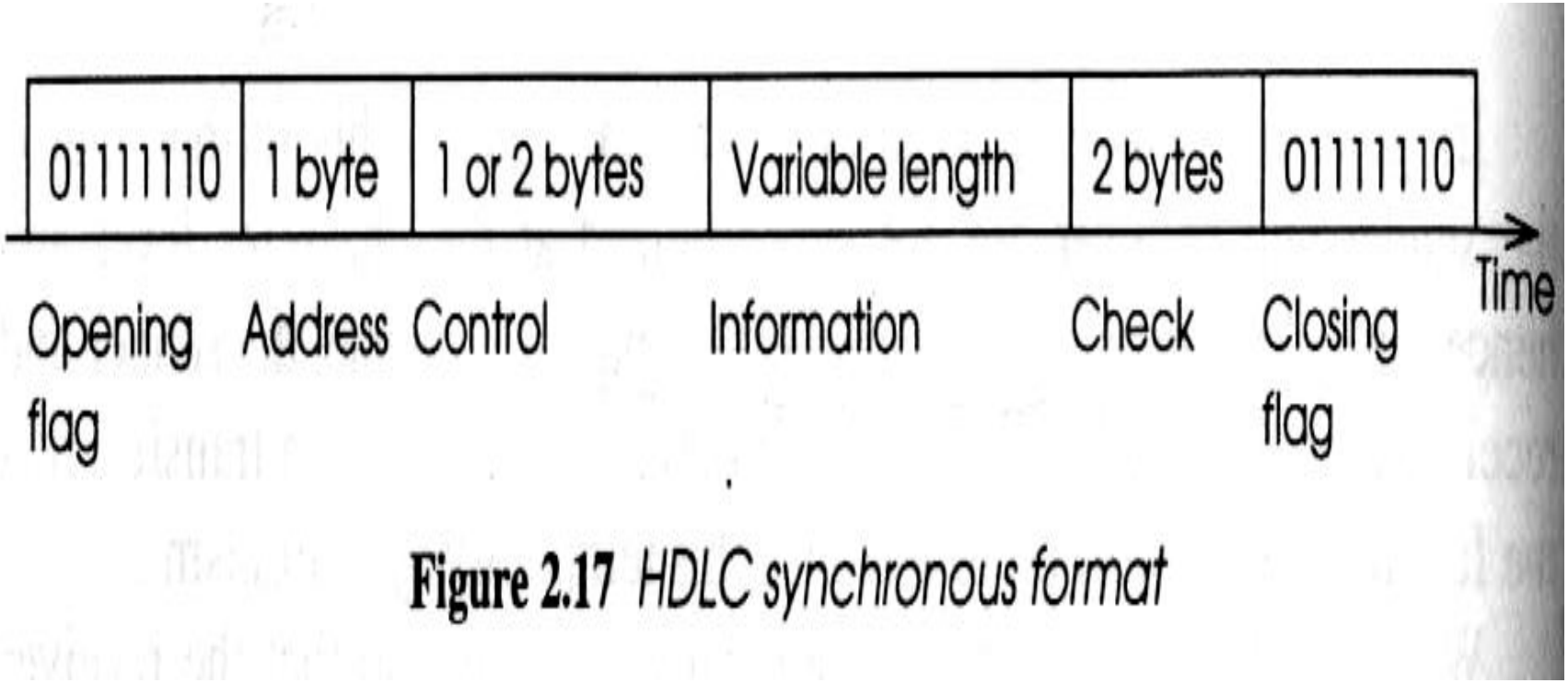


Figure 2.17 *HDLC synchronous format*

Synchronous Transmission: Bit Stuffing

- Character synchronization is achieved by identifying a special pattern of bits (01111110) known as a Flag.
- This pattern is unique and can never occur in the data stream except as a flag.
- Since the flag contains 6 consecutive 1s, data to be transmitted is checked for 5 consecutive 1s and if found a 0 is inserted after them. This technique is called *Bit Stuffing*.
- At receiving end, to removing these additional stuff bit and getting back the original data frame is called bit un-stuffing

Bit Stuffing

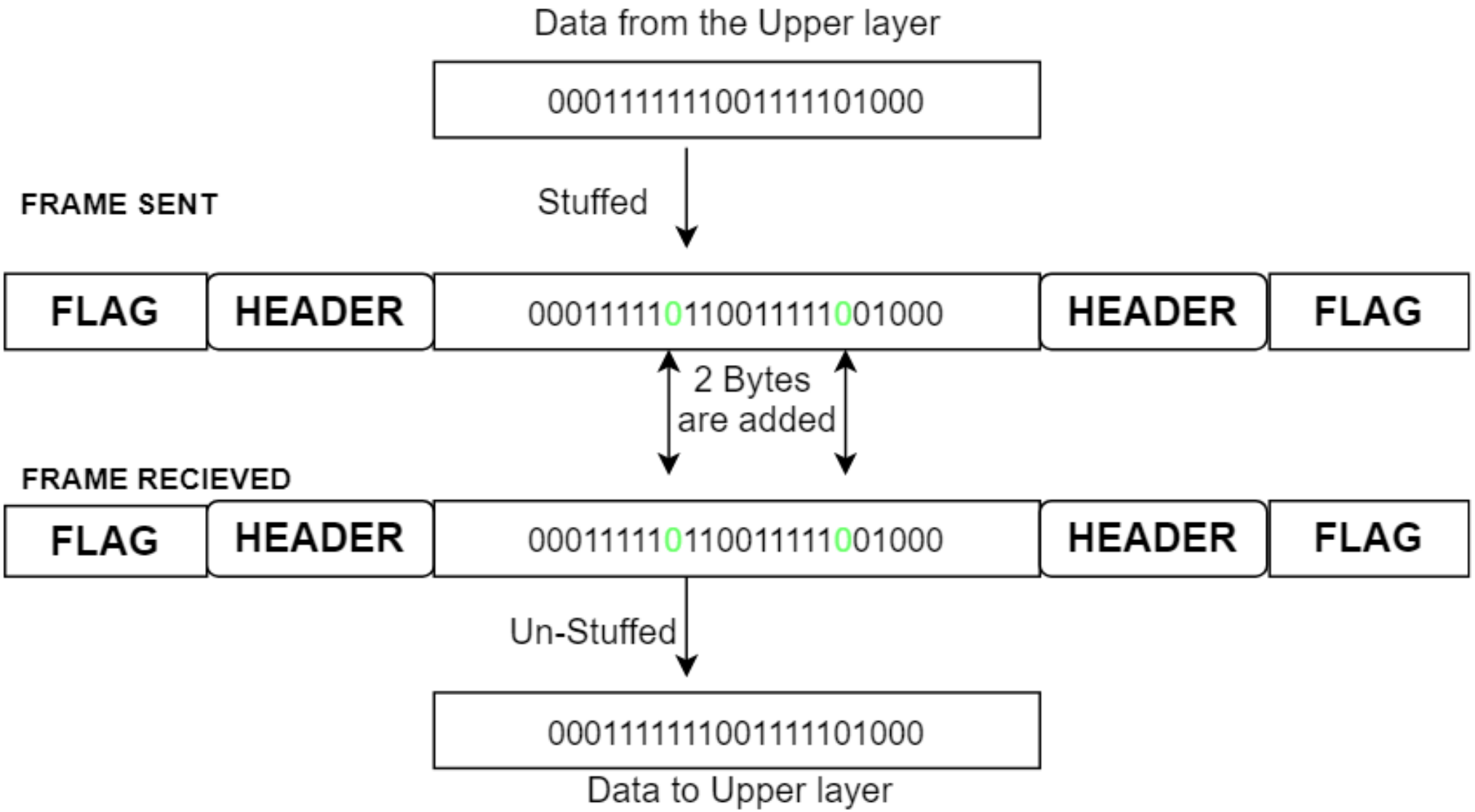
Definition: Bit Stuffing is the process of inserting non-information bits into data to break up bit patterns to affect the synchronous transmission of information. It is widely used in network and communication protocols, in which bit stuffing is a required part of the transmission process.

Each frame begins and ends with a special bit pattern called flag byte [01111110].

Whenever sender data link layer encounters five consecutive ones in the data stream, it automatically stuffs a 0 bit into the outgoing stream.

When the receiver sees five consecutive incoming ones, it automatically unstuffs the 0 bit before sending the data to the network layer.

Bit Stuffing and Bit Un-stuffing



Computer-Peripheral Connections

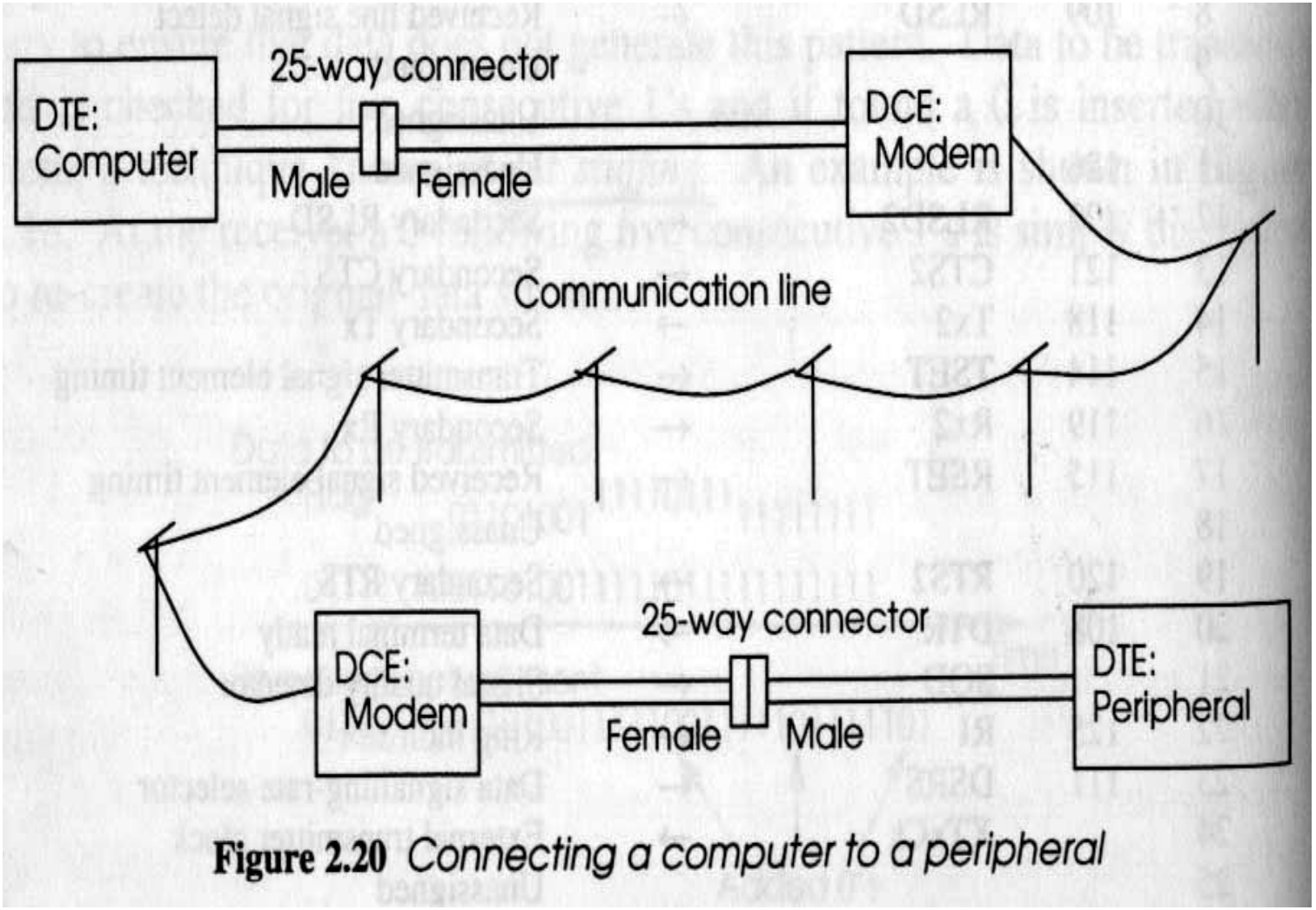


Figure 2.20 *Connecting a computer to a peripheral*